

Wydawnictwo Helion ul. Kościuszki 1c 44-100 Gliwice tel. 032 230 98 63 e-mail: helion@helion.pl



Tworzenie stron WWW. Ilustrowany przewodnik

Autor: Aleksandra Tomaszewska-Adamarek ISBN: 83-246-0608-4 Format: A5. stron: 184



Stwórz i opublikuj własną stronę WWW

- Wykorzystaj edytory wizualne
- Poznaj język HTML i JavaScript
- Umieść stronę w sieci

Trudno wyobrazić sobie współczesny świat bez internetu. Poczta elektroniczna, Gadu-Gadu, Skype, sklepy internetowe i portale są dziś równie powszechne jak kilkanaście lat temu radio i telewizja. Internet to potężne medium informacyjne i komunikacyjne. W dziesiatkach milionów witryn internetowych firmy prezentuja swoja ofertę, organizacje polityczne – swoje mniej lub bardziej niemożliwe do zrealizowania programy, a osoby prywatne - swoje zainteresowania, fotografie i coraz częściej opowieści o własnym życiu. W sieci można znaleźć firmy oferujące nieodpłatnie konta, na których można opublikować witrynę WWW. Tylko jak ją zrobić?

W książce "Tworzenie stron WWW. Ilustrowany przewodnik" znajdziesz odpowiedź na to pytanie. Czytając ją, poznasz wszystkie zagadnienia niezbędne do tego, aby Twoja witryna WWW pojawiła się w sieci. Dowiesz się, w jaki sposób stworzyć stronę WWW, korzystając z dostępnych w sieci narzędzi lub pisząc ją samodzielnie w języku HTML. Poznasz najpopularniejsze edytory służące do pisania kodu stron, nauczysz się przygotowywać grafikę na potrzeby sieci i rejestrować swoją witrynę w katalogach i wyszukiwarkach.

- Tworzenie własnego blogu
- Edytory kodu i narzędzia wizualne
- Przetwarzanie grafiki
- Elementy tekstowe
- Tworzenie formularzy
- Formatowanie stron za pomocą arkuszy stylów
- Tworzenie elementów dynamicznych w JavaScript
- Przygotowywanie animacji w programie Macromedia Flash

a saar le

Rejestracja witryny w wyszukiwarkach

Przekonaj się, że zaprojektowanie witryny WWW

Spis treści

Wstęp	6
Rozdział 1. Trzy, dwa, jeden… START! Blog Serwisy stron WWW	9 10 16
Rozdział 2. Edytory WYSIWYG, czyli chłopcy od brudnej roboty	22
Edycja kodu	34
Szablony	40
Zarządzanie szablonami	43
Rozdział 3. Grafika	
Dodawanie obrazków do strony	49
Tworzenie grafiki	54
Kadrowanie	56
Photoshop	56
Fireworks	56
Zmiana wielkości i rozdzielczości	57
Photoshop	57
Fireworks	57
Obracanie	58
Photoshop	58
Fireworks	58
Wyostrzanie i rozmazywanie	59
Photoshop	59
Fireworks	59
Stosowanie filtrów efektów artystycznych	61
Photoshop	61
Fireworks	61
Darmowa grafika	62
Tworzenie elementów rollover	64
Dzielenie obrazów na fragmenty	67

Rozdział 4. HTML – język Internetu	70
Podstawy i struktura dokumentu	71
Polecenia w nagłówku	73
Ciało dokumentu	75
Nagłówki	75
Akapity	76
Cytaty	76
Listy	77
Łącza	78
Tabele	80
Formularze	83
Pole INPUT	83
Pole SELECT	84
Pole TEXTAREA	84
Ramki	86
Adresowanie dokumentów wewnątrz ramek	
Dodatkowe znaczniki	
Pozdział 5. Co to znaczu mieć stul, czuli słów kilka o CSS	00
Formatowanie	
Pionowe i poziome wyrównanie tekstu	
Wciecie pierwszego wierszą akapitu	
Odsteny	
Dekoracja tekstu	
Małe i wielkie litery	100
Formatowanie wydladu czcionki	100
Kolor i tło	104
Marginesy	
Programy do tworzenia arkuszy stylów	109
Rozdział 6. Tworzenie dynamicznych stron WWW,	
czyli JavaScript w akcji	111
Typy danych	114
Boolean	114
Number	114
String	114
Null i Undefined	115
Wyrażenia i warunki	116
Inne operatory	118
Wprowadzanie poleceń	119

Wprowadzanie poleceń w elemencie SCRIPT	
Łaczenie poleceń w funkcie	
Sterowanie przepływem	
ifelse	
while	
dowhile	
for	
break i continue	
switch	
Okna	
Tworzenie nowego okna przeglądarki	
Uruchamianie skryptów za pomocą zdarzeń	
Dezdziel 7. Animacia Elech	126
Rozuziar 7. Animacja Flash	100
Przygolowywanie granki	، 120
Wypołnionio	1/11
	۱ ۹ ۱
	144
Skalowanie i obracanie	1/5
Przekształcenia numeryczne	146 146
Taket	140 146
Animacie	140 147
Przesuwanie grafiki no ścieżce	
Dźwiek	156
Eksportowanie i publikowanie filmów	
Wskazówki pomocne w optymalizowaniu animacii Flash	166
Rozdział 8. Promocja	
Wyszukiwarki internetowe	
Rejestracja	
Element IIILE	
Reklama	
Katalogi internetowe	
Ogłoszenia prasowe	
Poczta elektroniczna	
Banery reklamowe	
Agencje reklamowe	
Skorowidz	

Rozdział 5. Co to znaczy mieć styl, czyli słów kilka o CSS

rzyciski, paski menu, zdjęcia, obrazy - nieodłaczne cześci stron WWW. Wiesz już, jak je tworzyć i jak budować z nich całość za pomocą edytorów tekstowych oraz programów WYSIWYG. Znasz podstawy języka HTML, a teraz nadszedł czas, aby wyjaśnić, co to są te kaskadowe arkusze stylów, które wielokrotnie pojawiały się w poprzednich rozdziałach. W Paryżu lub Nowym Jorku określenie "mieć styl" oznacza, że chodzisz w butach od Gucciego, marynarkach od Armaniego i dzinsach od Marca Jacobsa. Na szczeście w wirtualnym świecie styl kosztuje dużo mniej. Aby mieć styl, wystarczy się zapoznać ze specyfikacją kaskadowych arkuszy stylów i stosować tę technologię do budowy tworzonych stron.

Kaskadowe arkusze stylów (ang. cascading style sheets — CSS) są stosowane do formatowania stron i pozwalają uzyskać doskonałą kontrolę nad wyglądem całego dokumentu. Sensowność korzystania z kaskadowych arkuszy stylów nietrudno udowodnić — wyobraź sobie witrynę składającą się z dużej liczby stron WWW, nad którą pracuje cała grupa programistów. Mając już wyobrażenie o stopniu skomplikowania zadania polegającego na stworzeniu kilku stron, na których używana będzie ta sama stylistyka — krój, kolor, wielkość czcionki, kolorystyka elementów czy wygląd tabel, wyobraź sobie, jak trudno jest zapanować nad tymi wszystkimi elementami, gdy masz do ogarnięcia kilkaset stron projektowanych przez kilka różnych osób.

Korzystanie z kaskadowych arkuszy stylów oznacza również oszczędność miejsca, ponieważ styl każdego elementu jest definiowany jednokrotnie w pliku definicji stylów, a nie przy każdym wystąpieniu elementu na stronie, czyli w przypadku takich elementów jak na przykład akapit czy nagłówek kilkadziesiąt razy na każdej stronie.



Wskazówka

Szczegółowe informacje na temat kaskadowych arkuszy stylów znajdziesz pod adresem: http://www.w3.org/TR/REC-CSS 2 Definicje stylów można wprowadzać do opisu strony na kilka sposobów, które teraz omówię. Pamiętaj, że każdy z tych sposobów ma swoje wady i zalety, a wybór najlepszego sposobu zależy od preferencji programisty oraz sytuacji, w której styl ma być zastosowany.

Sposób 1.

Zawartość
akapitu itd. itd. itd.

Definicja stylu jest umieszczona bezpośrednio w znaczniku. Zaletą tego rozwiązania jest prostota jego użycia. Styl jest umieszczony wewnątrz znacznika tak, jak umieściłbyś tam odpowiednie atrybuty znacznika. Wadą jest konieczność powtarzania tej definicji dla każdego kolejnego znacznika umieszczanego na stronie.

Sposób 2.

```
<html>
<head>
<style type="text/css">
p
{
color: blue;
font-family: Arial;
font-size: 12pt;
}
</style>
</head>
```

<body>

Pierwszy akapitDrugi akapit</body>

</html>

Definicja stylu jest umieszczona w nagłówku dokumentu. Tak zdefiniowany styl odnosi się do wszystkich znaczników tego typu w dokumencie. W naszym przykładzie styl dotyczy wszystkich akapitów, które zostaną umieszczone w dokumencie. Ale jeśli znajdą się w nim również nagłówki, obrazy czy tabele, to definicja stylu nie będzie się już do nich odnosić — będą one wyświetlane zgodnie z domyślnymi wartościami. Takie rozwiązanie pozwala na jednokrotną definicję stylu dla elementu w obrębie strony. Jeśli jednak zechcesz użyć tego samego stylu na innych stronach, to musisz go ponownie zdefiniować w nagłówkach tych stron.

```
Sposób 3.
```

<html></html>
<head></head>
<link <br="" href="style.css"/> rel="stylesheet" type="text/css">
<body></body>
Pierwszy akapit
Drugi akapit

Trzecim sposobem jest umieszczenie definicji stylów w osobnym pliku o rozszerzeniu *.css, który jest powiązany z dokumentem strony poprzez użycie znacznika <LINK>. Atrybut href wskazuje ścieżkę dostępu do pliku arkuszy stylów. Wywołanie zewnętrznego arkusza stylów znajduje się w nagłówku i dotyczy całej strony WWW, czyli wszystkich elementów danego typu.

W pliku zewnętrznego arkusza stylów znajdują się wyłącznie wpisy definiujące style określonych elementów. Do tak zdefiniowanego arkusza można podłączyć wiele stron witryny i wszystkie będą miały zdefiniowane te same style. Dzięki temu w jednym miejscu możesz definiować i zarządzać wyglądem całej witryny.

Arkusz stylów może mieć w tym wypadku postać:

```
    y danego typu elementu dla wszystkich
    e stron powiązanych z arkuszem.
    e Filozofia kaskadowych arkuszy stylów
```

łączy w sobie elastyczny sposób kontroli stylów stron WWW i poszczególnych elementów na stronach z odpowiednią hierarchią stylów, która pozwala unikać konfliktów, jeśli style elementów są definiowane na kilku poziomach. Sposób 1. (atrybut style) ma najwyższą rangę w hierarchii, sposób 2. (element STYLE) ma wyższą rangę niż styl definiowany przez zewnętrzny arkusz stylów, ale niższą niż atrybut style. Jak łatwo się domyślić, najniższą rangę ma styl przypisywany przez zewnętrzny arkusz stylów. Takie rozwiązanie pozwala uniknąć konfliktów, które

za pomocą zewnętrznego arkusza

stylów - styl jest definiowany dla

```
02
```

Wskazówka

Pamiętaj, że zarówno element <STYLE>, jak i element <LINK> muszą być umieszczone wewnątrz nagłówka strony.

{ color: blue; font-family: Arial; font-size: 12pt; }

р

Kaskadowe arkusze stylów nieprzypadkowo noszą taką nazwę, bo jak wiesz, style można wprowadzać na trzech różnych poziomach:

- za pomocą atrybutu style (sposób 1.)
 styl definiowany kolejno dla poszczególnych elementów na stronie;
- za pomocą elementu STYLE (sposób 2.)
 styl elementu jest określany dla całej strony;

atrybut style

element STYLE

zewnętrzny arkusz stylów

mogłyby wyniknąć, jeśli na stronie użylibyśmy wszystkich trzech metod definiowania stylów. Hierarchia — struktura stylów — przypomina stopnie wodospadu, stąd bierze się w nazwie określenie *kaskadowy*.

Hierarchizacja stylów obejmuje nie tylko różne sposoby definiowania stylów, ale również sytuację, gdy do jednej strony podłączonych jest kilka różnych zewnętrznych arkuszy stylów. Przeglądarka uwzględnia w takim wypadku kolejność wprowadzania stylów. Zewnętrzny arkusz wprowadzony jako pierwszy dominuje nad arkuszem, który jest wprowadzony jako następny w kolejności. Problem polega na tym, że jeśli w zewnętrznych arkuszach stylów występują wzajemne konflikty, trudno przewidzieć efekt graficzny, dopóki strona nie zostanie wyświetlona w przeglądarce.

Do określenia wartości atrybutów arkusze stylów wykorzystują względne i bezwzględne jednostki miary oraz określone definicje kolorów.

Jednostki względne:

- px piksele jednostka definiowana w oparciu o pojedyncze punkty świetlne monitora;
- em określa zależność pomiędzy wysokością elementu a wielkością zdefiniowanej czcionki;
- ex proporcje do wysokości litery;
- % procenty określenie wielkości względem wartości domyślnej.



Wskazówka

Istnieje także możliwość importowania arkusza stylów wewnątrz innego arkusza stylów. W tym celu należy użyć następującej konstrukcji:

@import url(style.css) ;
body {background-image: url(obraz.
gif) ;}

W takiej sytuacji importowany arkusz ma niższy priorytet niż arkusz, do którego import następuje.

Jednostka em jest jednostką umowną, która jest równa innej zdefiniowanej wartości. Jeśli ustalimy wielkość czcionki na 12 pt, to 1em będzie równy właśnie tej ustalonej wartości, czyli 12 pt. Jeśli określę inny atrybut jako 2 em, będzie on dwukrotnością wcześniej zdefiniowanej wartości (czyli w tym wypadku będzie to 24 pt).

Jednostki bezwzględne:

in — cale — jednostka najczęściej używana w USA, rzadko używana w Polsce (1 in = 2,54 cm);

pt — punkty — wywodzą się z typografii, gdzie są standardową jednostką miary (72 pt = 1 in);

cm — centymetry — jednostka pochodna systemu metrycznego, często stosowana w Europie;

mm — milimetry — jednostka pochodna systemu metrycznego, często stosowana w Europie;

pc — pica — jednostka wywodząca się z typografii (1 pica = 12 pt). W kaskadowych arkuszach stylów można definiować kolory poprzez użycie ich nazw angielskich, jak blue czy yellow. Wszystkie kolory, które nie zostały zdefiniowane przez użycie ich nazw, muszą być określone przez wykorzystanie modelu RGB. Polega on na przypisaniu liczb całkowitych z przedziału od 0 do 255 każdej z wartości składowych palety RGB. Na przykład color: rgb(0,12,255). Kolor można również zapisać, definiując procent nasycenia danej barwy, na przykład color: rgb(10%, 50%, 0%).

Kolor można również zapisać, stosując metodę szesnastkową, na przykład color: #ff2c00.

black	#000000
green	#008000
silver	#C0C0C0
lime	#00FF00
gray	#808080
olive	#808000
white	#FFFFFF
yellow	#FFFF00
maroon	#800000
navy	#000080
red	#FF0000
blue	#0000FF
purple	#800080
teal	#008080
fuchsia	#FF00FF
aqua	#00FFFF

Schemat konstrukcji stylu można zapisać w następujący sposób:

selektor {cecha: wartość; cecha: wartość; itd.}

- selektor znacznik języka HTML, np. p to paragraf, h1 — nagłówek pierwszy, a table — tabela.
- cecha wpis określony specyfikacją kaskadowych arkuszy stylów. Przykładowe właściwości określone przez specyfikację kaskadowych arkuszy stylów to background-color, border-style, font-family itd. Wartość jest zazwyczaj przypisana do danej cechy, ale może się zdarzyć, że będzie podobna dla kilku zupełnie różnych cech.

Ogólnie przyjmujemy, że selektor to dowolny znacznik języka HTML. Możemy jednak wyróżnić kilka rodzajów selektorów:

- Selektory proste selektor będący pojedynczym znacznikiem języka HTML, odnoszący się jedynie do tego elementu. Jest to najczęściej spotykany rodzaj selektora.
- Selektor uniwersalny używany, gdy jakieś cechy i wartości są uniwersalne i powinny dotyczyć wszystkich elementów na stronie.
- Potomek selektor zbudowany w oparciu o zależności panujące pomiędzy poszczególnymi znacznikami języka HTML.

Przykłady selektorów uniwersalnych:

```
*
{color: blue;}
BODY
{color: blue}
```

Tak zapisane selektory działają dla wszystkich elementów strony — nagłówki, akapity, listy itd. będą miały kolor niebieski.

Przykłady potomków:

```
p
{color: blue}
p > span
{font-style: italic}
```

Kolor tekstu akapitu będzie niebieski, ale dodatkowo, jeśli wewnątrz akapitu pojawi się znacznik , to otoczony nim tekst będzie pochylony.

Specyfikacja kaskadowych arkuszy stylów dopuszcza również możliwość grupowania selektorów. Jeśli na przykład chciałbyś, aby wszystkie listy, które wystąpią w dokumencie, miały kolor czerwony, zamiast definiować styl dla każdego typu listy możesz zastosować następujący zapis:

```
ul, ol, dl
{
color: red
}
```

Selektory ID pozwalają określić styl danego elementu na stronie, co pozwala zróżnicować formatowanie elementów tego samego typu na stronie. Wyobraź sobie, że wszystkie akapity na stronie mają mieć kolor granatowy, ale w jednym tekst powinien być czerwony, bo zawiera on szczególnie ważne informacje. W tym celu możesz zdefiniować style w następujący sposób:

p {color: navy}
p#wazne {color: red}

Aby utworzyć akapit zawierający tekst w kolorze czerwonym, musisz go wywołać przez użycie zdefiniowanej nazwy:

 Bardzo ważne informacje wyróżnione kolorem czerwonym

Pozostałe akapity będziesz tworzyć, używając znacznika .

Definicja identyfikatora może również mieć następującą postać:

#duzy_czerwony_tekst

{font-size: 20pt; color: red}

Identyfikator ten możesz wywołać w dowolnym znaczniku, korzystając z atrybutu ID.



Deklaracja selektora ID musi być poprzedzona znakiem #, ale znaku tego nie używa się do odwoływania się do wartości atrybutu ID.

Kolejnym podstawowym elementem specyfikacji kaskadowych arkuszy stylów jest klasa. Klasę konstruuje się w następujący sposób:

selektor.nazwa_klasy {cecha: wartość}

Tak zdefiniowana klasa może być wywołana tylko w znaczniku, dla którego została utworzona.

Aby wywołać klasę w dokumencie, należy użyć atrybutu CLASS. Na przykład tworząc klasę akapitu zawierającego tekst w kolorze czerwonym, należy wpisać definicję:

p.czerwony {color: red}

Uwaga

Aby wywołać tę klasę w dokumencie, wpisz:

Czerwony tekst



Pamiętaj, aby definiując klasę, umieścić znak kropki między selektorem a nazwą klasy.

Dziedziczenie i kaskadowość to cechy, którym kaskadowe arkusze stylów zawdzięczają swój dynamiczny rozwój i pozycję. O kaskadowości rozmawialiśmy już wcześniej w tym rozdziale, a teraz warto wspomnieć kilka słów o dziedziczeniu.

Jeśli chcesz określić kolor tekstu w komórce tabeli, zrobisz to przez użycie elementu TD. Możesz to również zrobić przez użycie elementu TABLE i wtedy wszystkie komórki tabeli, zarówno w wierszach, jak i w nagłówkach, będą miały określony dla tabeli kolor. Ale możesz to również zrobić, definiując kolor tekstu dla elementu BODY. Wtedy nie tylko tekst w komórkach tabeli będzie miał określony kolor, ale również tekst w akapitach, nagłówkach, listach będzie miał określony kolor. Na tym właśnie polega dziedziczenie w kaskadowych arkuszach stylów — elementy nadrzędne domyślnie przekazują zdefiniowane właściwości elementom, które znajdują się niżej w hierarchii.

W naszym przykładzie kolor niebieski dla tekstu zdefiniowany w elemencie BODY będzie oznaczał również niebieski kolor tekstu we wszystkich komórkach tabeli. Jeśli jednak określimy atrybut elementu podrzędnego, to nie przekaże on tej właściwości w górę hierarchii. W naszym przykładzie niebieski tekst zdefiniowany dla komórki tabeli nie będzie oznaczał niebieskiego tekstu dla wszystkich elementów wewnątrz elementu BODY.



Dziedziczenie działa domyślnie, czyli dla tych elementów, dla których nie określono innej wartości właściwości niż ta, którą wskazano dla elementu nadrzędnego. Możemy zdefiniować czerwony tekst w komórkach tabeli, mimo że dla elementu BODY zdefiniowano, że tekst ma być niebieski.

Większość ludzi ogranicza wykorzystanie kaskadowych arkuszy stylów do określania koloru tekstu w dokumencie lub kroju czcionki w nagłówkach. Jednak istnieją również ciekawsze zastosowania kaskadowych arkuszy stylów — mogą one służyć do tworzenia menu, do którego większość projektantów zaangażowałaby JavaScript.



<html>

```
<head><link rel="stylesheet"
href="menu.css" type="text/css">
</head>
<body>
Firma
```

Menu utworzone bez użycia JavaScriptu, a wyłącznie z wykorzystaniem kaskadowych arkuszy stylów

```
</body>
```

```
</html>
```

menu.css:

body{background-color: lightblue; font-size: larger}

#menu { background-color: blue;
float: left }

#menu li {font-size: x-large; color: yellow}

#menu li a {color: lime}

#menu li a hover {color: navy; background-color: white; font-size: larger }

Formatowanie

Pionowe i poziome wyrównanie tekstu

Specyfikacja kaskadowych arkuszy stylów pozwala na kontrolowanie następujących właściwości tekstu:

- wyrównanie poziome i pionowe,
- wcięcie,
- odstępy między wierszami,
- odstępy między wyrazami,
- odstępy między literami,
- dekoracja tekstu,
- przekształcanie tekstu,
- kontrola pustej przestrzeni.
- do prawej text-align: left,
- do lewej text-align: right,
- wyśrodkowanie text-align: center,
- wyjustowanie bloku tekstu textalign: justify,
- do góry vertical-align: top,
- do środka vertical-align:middle,
- do dołu vertical-align: bottom,
- indeks dolny vertical-align: sub,
- indeks górny vetical-align: super.

Wcięcie pierwszego wiersza akapitu

Aby uzyskać efekt wcięcia pierwszego wiersza w akapicie, należy użyć polecenia text-indent i określić głębokość tego wcięcia w jednostkach względnych lub bezwzględnych. Na przykład:

p {text-indent: 1 cm}



Odstępy

Za sterowanie odstępem pomiędzy wierszami tekstu odpowiada polecenie lineheight, które może przyjmować wartości dodatnie i ujemne. Przy użyciu wartości dodatnich odstępy między wierszami się zwiększają, a gdy użyjesz wartości ujemnych, tekst będzie się zagęszczał.

ЯК	askadow	e arku	sze stylów	/ Micros	oft Internet	l Explorer		
BK	Edycja	widok	Ulubione	Narzędzia	Pomos			1
Jest	teśmy d	ynamicz	mie rozwi	jającą się	firmą, która	a działa na p	olskim	
ryni	ku od 1	997 rok	cu. 10 lat	na polskin	n rynku to (dbagi czas, al	e 10 lat ni	i.
pol	skim ryr	sku info	rmatyczny	rm to dośr	wiadczenie	i profesjonal	izm, które	:
2.				2				

Kaskadowe arkusze stylów pozwalają również definiować odstępy pomiędzy poszczególnymi wyrazami tekstu. Właściwość ta jest określana przez parametr word-spacing. Również i w tym wypadku możliwe jest podanie wartości dodatnich (zwiększenie odstępu między wyrazami) i ujemnych (zmniejszenie odstępu między wyrazami).



Twórcy kaskadowych arkuszy stylów poszli jeszcze o krok dalej i pozwolili projektantom stron wpływać nawet na odstęp liter w tekście. Efekt ten można uzyskać korzystając z polecenia letterspacing.

Na przykład:

P {letter-spacing: 5pt}



Dekoracja tekstu

Używając kaskadowych arkuszy stylów, twórcy stron mogą kontrolować wygląd tekstu za pomocą właściwości text-decoration. Korzystając z tego parametru, mogą oni sprawić, że tekst będzie:

podkreślony — text-decoration: underline,

przekreślony — text-decoration: linethrough,

nadkreślony — text-decoration: overline,

migający — text-decoration: blink,

nie będzie dekoracji tekstu — text-decoration: none.

Korzystając z tego parametru, możesz na przykład wyłączyć podkreślenia wszystkich łączy na stronie:

A {text-decoration: none;}



Małe i wielkie litery

Nie trudząc się zbytnio, możesz zamienić cały tekst na stronie na wielkie litery, chociaż oryginalnie w pliku jest on wpisany małymi literami. Na taką wygodę pozwala parametr text-transform. Może on przyjmować następujące wartości:

- tylko duże litery text-transform: uppercase,
- tylko małe litery text-transform: lowercase,
- każda pierwsza litera w słowie jest zamieniana na wielką — text-transform: capitalize,
- bez zmian text-transform: none.

Zamiast wstawiać do dokumentu twarde spacje, możesz użyć parametru white-space, który przyjmuje następujące wartości:

- normal spacje są redukowane do jednej,
- pre zachowanie dokładnej liczby spacji znajdujących się w tekście,
- nowrap blokuje automatyczne łamanie wierszy.



Formatowanie wyglądu czcionki

Zgodnie ze specyfikacją kaskadowych arkuszy stylów możliwe jest definiowane następujących własności czcionki:

- rodzaju atrybut font-family,
- rozmiaru atrybut font-size,
- wagi atrybut font-weight,
- stylu atrybut font-style.

Pod pojęciem rodzaju czcionki kryje się rozróżnienie pomiędzy krojami czcionek lub rodzinami czcionek. Za pomocą kaskadowych arkuszy stylów możesz określić rodzaj czcionki, z którego chcesz korzystać przy wyświetlaniu wskazanego fragmentu tekstu. Konstrukcja stylu jest budowana według następującej definicji:

selektor {font-family: nazwa własna czcionki lub nazwa rodzajowa czcionki}

nazwa własna czcionki — np. Arial, Times New Roman, Verdana; nazwa rodzajowa czcionki — jedna z pięciu wartości: serif, sans-serif, monospace, cursive lub fantasy.

Czcionki szeryfowe (serif) charakteryzują się tym, że każda litera posiada ozdobniki zwane szeryfami. Dodatkowo czcionki tego typu są proporcjonalne, gdyż każda litera ma inną szerokość. Typowym przedstawicielem tej grupy czcionek jest czcionka Times New Roman.

Przykład czcionki szeryfowej (serif)

a b c d e f g h i j k l m n O P Q R S T U W X Y Z

100

Czcionki bezszeryfowe nie mają żadnych ozdobników na czcionce, ale również są proporcjonalne. Ten rodzaj czcionek jest najczęściej używany do tworzenia tekstu przeznaczonego do czytania (treść strony). Typowym przedstawicielem tej grupy czcionek jest czcionka Arial.

Przykład czcionki bezszeryfowej (sans-serif)

a b c d e f g h i j k l m N O P Q R S T U W X Y Z

Czcionki tego typu są nieproporcjonalne, co oznacza, że każda litera ma taką samą szerokość. Ten rodzaj czcionek automatycznie kojarzy się z tekstem napisanym na maszynie do pisania. Typowym przedstawicielem tej grupy czcionek jest czcionka Courier New.

Przykład czcionki typu monospace

а	b	С	d	е	f	g	h	i	j
k	1	m							
Ν	0	Ρ	Q	R	S	Т	U	W	Х
Y	Ζ								

Czcionki tego typu charakteryzują się dużym podobieństwem do pisma odręcznego, mogą też zawierać różnego rodzaju ozdobniki w postaci zawijasów. Przykładem czcionki typu cursive może być czcionka Comic Sans MS.

Przykład czcionki typu cursive

abcdefghijklm NOPQRSTUWXY Z

Czcionki typu fantasy to czcionki, których nie da się zakwalifikować do żadnego z opisanych powyżej rodzajów czcionek.

Przykład czcionki typu fantasy



Uwaga

Należy pamiętać, że czcionki zdefiniowane w arkuszu stylów są pobierane z komputera użytkownika przeglądającego stronę. Jeśli w systemie nie ma zainstalowanej czcionki określonej przez arkusz stylów, przeglądarka zastąpi ją domyślną czcionką dla danego systemu. Dlatego przy projektowaniu stron nie należy używać zbyt wymyślnych i rzadko spotykanych krojów czcionek. Warto trzymać się najpopularniejszych, jak Times, Arial czy Helvetica.

Za rozmiar czcionki użytej na stronie odpowiada parametr font-size. Rozmiar czcionki może być podany w jednostkach względnych lub bezwzględnych. Możesz w związku z tym używać następujących definicji rozmiarów czcionki:

- font-size: 20 pt,
- font-size: 12 px,
- font-size: 200%.

Ciekawym rozwiązaniem jest również możliwość korzystania ze zdefiniowanych wielkości: xx-small, x-small, medium, large, x-large, xx-large. Dodatkowo istnieje również możliwość określenia względnych rozmiarów za pomocą opcji larger oraz smaller. Na przykład:

```
BODY
{font-size: large;}
.wiekszy
{ font-size: larger;}
.mniejszy
```

```
{font-size: smaller;}
```



Pojęcie wagi czcionki odnosi się do jej grubości, a do definiowania grubości czcionki używany jest atrybut fontweight. Wartość może być określona za pomocą liczb z przedziału od 100 do 900 oraz nazw bold, bolder, lighter i normal. Styl czcionki jest definiowany przez parametr font-style, który może przyjmować jedną z dwóch wartości — italic (kursywa) lub obligue.



Kontrola nad wyglądem list za pomocą kaskadowych arkuszy stylów polega na sterowaniu właściwościami użytej czcionki oraz określaniu typu listy, użytego punktora graficznego oraz definiowaniu pozycji elementów listy względem punktora.

Typy list dzielimy na punktowane, numerowane i definicje. Od typu listy uzależnione są rodzaje punktorów, których możemy użyć w liście. Dla list numerowanych punktorami mogą być litery (małe lub wielkie) oraz cyfry (arabskie lub rzymskie). Dla list punktowanych możemy użyć takich punktorów jak kwadrat, koło lub okrąg.

Typ listy określa właściwość list-style-type. Dla listy punktowanej może on przyjmować wartości:

- disc punktor ma postać koła;
- circle punktor ma postać okręgu;
- square punktor ma postać kwadratu.

Dla listy numerowanej właściwość liststyle-type może przyjmować wartości:

- decimal odpowiada liczbom arabskim;
- lower-roman odpowiada małym liczbom rzymskim;
- upper-roman odpowiada dużym liczbom rzymskim;
- lower-alpha odpowiada małym literom;
- upper-alpha odpowiada dużym literom;
- none brak wypunktowania.



Czasem projektanci chcą zabłysnąć zupełnie nową i niestandardową koncepcją listy, i właśnie na takie okazje twórcy kaskadowych arkuszy stylów przygotowali parametr list-style-image, który pozwala użyć jako punktora listy wskazanego pliku graficznego w formacie GIF, JPG lub PNG.

Należy jednak pamiętać, że obrazek nie powinien być zbyt duży, gdyż będzie psuł efekt listy. Czasem może również wystąpić konieczność regulacji położenia tekstu względem punktora. Do tego celu służy parametr list-style-position.

Parametr list-style-position przyjmuje następujące wartości:

- outside punktor znajduje się wyraźnie poza listą;
- inside punktor jest schowany w tekście listy.



Jeżeli lista zawiera krótkie wpisy w każdym punkcie, to różnica pomiędzy ustawieniami parametru list-style-position będzie dla użytkownika trudna do zauważenia. Stosowanie tego parametru jest zasadne, jeśli każda pozycja listy składa się z więcej niż jednego wiersza.



Kolor i tło

Każdy element strony, który możesz sformatować przez użycie kaskadowych arkuszy stylów, może mieć zdefiniowany kolor i tło. Jako tło elementy mogą mieć zdefiniowany kolor lub użycie wskazanego obrazu.

Aby przypisać kolor do elementu strony, należy użyć parametru color i podać wartość koloru przez użycie jego zwyczajowej nazwy angielskiej, zapisu palety rgb lub określenie wartości szesnastkowej koloru. Szczegółowe sposoby definiowania kolorów znajdziesz wcześniej w tym rozdziale.

Tło elementu możesz określić poprzez podanie koloru tła:

selektor {background-color: wartość;}

lub wskazanie obrazu, który ma zostać użyty jako tło obiektu:

```
selektor {background-image: URL
(obraz.gif)}
```

Obraz użyty jako tło elementu strony musi być zapisany w formacie JPG, GIF lub PNG.

Jeśli rozmiar obrazu użytego jako tło elementu jest mniejszy niż rozmiar tego elementu, to obraz w tle będzie powielany, aby wypełnić całkowicie tło. Jednak za pomocą kaskadowych arkuszy stylów możliwe jest wprowadzenie kontroli nad powielaniem oraz zablokowanie przewijania tła. Za powielanie obrazu w obrębie elementu, dla którego zdefiniowane jest tło będące obrazem, odpowiada parametr background-repeat, który może przyjmować następujące wartości:

- no-repeat tło nie będzie powielane i zostanie wyświetlone tylko jeden raz;
- repeat-x tło będzie powielane wyłączenie w poziomie;
- repeat-y tło będzie powielane wyłącznie w pionie;
- repeat tło będzie powielane w pionie i w poziomie.



Dodatkowo istnieje możliwość zatrzymania tła, aby nie przesuwało się wraz z zawartością całej strony. Do tego celu używa się parametru background-attachment z ustawioną wartością fixed. Na przykład:

```
BODY
```

```
{background-image: url(tlo.gif);
background-attachment: fixed;
```

```
}
```

Specyfikacja kaskadowych arkuszy stylów pozwala nie tylko zdefiniować sposób powtarzania obrazu umieszczonego w tle elementu, ale również pozwala określić pozycję obrazu względem okna przeglądarki lub jego lewego górnego rogu. Właściwość tę określa parametr background-position, który może przyjmować jedną z następujących wartości:

- center tło wyśrodkowane,
- Ieft tło z lewej strony okna,
- right tło z prawej strony okna,
- top tło znajduje się na górze okna,
- bottom tło znajduje się na dole okna,
- wartość liczbowa określa odległość tła od lewego górnego rogu okna przeglądarki.

Wartości parametrów można ze sobą łączyć, dzięki czemu można na przykład umieścić tło w prawym górnym rogu okna przeglądarki, stosując zapis

background-position: right top



Jeśli zdecydowałeś się na użycie wartości liczbowych do określenia położenia tła, możesz stosować wartości względne lub bezwzględne.

Szczegółowy opis wartości względnych i bezwzględnych stosowanych w kaskadowych arkuszach stylów znajdziesz wcześniej w tym rozdziale.



Marginesy

Podczas tworzenia stron WWW można definiować dwa rodzaje marginesów: zewnętrzne i wewnętrzne. Marginesy zewnętrzne strony to margines lewy, prawy, górny i dolny. Do określania szerokości marginesu należy użyć parametru margin w połączeniu z określeniem położenia marginesu:

- margin-left lewy margines,
- margin-right prawy margines,
- margin-top górny margines,
- margin-bottom dolny margines.

Parametr przyjmuje wartości liczbowe względne i bezwzględne. Marginesy są definiowane w sposób niezależny od siebie (górny może być inny niż dolny, a prawy może być inny niż lewy) i mogą przyjmować wartości ujemne.



Uwaga

Pamiętaj, że marginesy różnych elementów się sumują. Jeśli na przykład ustawisz lewy margines dla elementu BODY na 2 cm, a następnie umieścisz na stronie tabelę, dla której ustawisz lewy margines na 2 cm, to lewa krawędź tabeli zostanie odsunięta od lewej krawędzi strony o 4 cm.

Przykładowa definicja marginesów na stronie:

BODY

{

```
margin-left: 1 cm;
margin-right: 2 cm;
margin-top: 1 cm;
margin-bottom: 3 cm;
}
```

Margines wewnętrzny jest definiowany za pomocą parametru padding w połączeniu z określeniem strony, której dotyczy margines. Określenie strony należy podać w języku angielskim: right (prawy), left (lewy), top (górny) i bottom (dolny).



Wskazówka

Jeśli chcesz ustawić identyczne wartości dla wszystkich marginesów, możesz to zrobić, używając wyłącznie parametru margin, na przykład: BODY

{ margin: 12 pt }



Kaskadowe arkusze stylów pozwalają pozycjonować na stronie dowolne elementy. Możesz to robić na trzy sposoby:

- bezwzględnie,
- względnie,
- statycznie.

Pozycjonowanie bezwzględne polega na określaniu dokładnego położenia elementów względem lewego górnego rogu okna przeglądarki. Przyjmuje się, że lewy górny róg okna przeglądarki ma współrzędne (0, 0). Elementy na stronie pozycjonuje się przez podanie wartości trzech parametrów: position, left i top. Przykład pozycjonowania bezwzględnego tabeli:

```
TABLE
```

{
position: absolute;
left: 100px;
top: 150px;

```
}
```



Pozycjonowanie względne definiuje się również przez podanie wartości dla parametrów left i top, ale właściwość position przyjmuje wartość relative. Pozycjonowanie względne odnosi się do domyślnego położenia elementu na stronie.

Przykład definicji stylu dla pozycjonowania względnego:

DIV
{
position: absolute;
top: 20px;
left: 20px;
}

Dla wybranych elementów można również zdefiniować ich wysokość i szerokość. Do tego celu w kaskadowych arkuszach stylów służą parametry width i height. Korzystając z pozycjonowania i wymiarowania, możesz stworzyć dwa akapity tekstu, które będą umieszczone obok siebie i w określonym miejscu na stronie.



Przykład wymiarowania elementów za pomocą kaskadowych arkuszy stylów:

```
<html>
<head>
<style type="text/css">
p.pierwszy
{position: absolute;
left: 20px;
top: 20 px;
width: 200px;
}
p.drugi
{position: absolute;
left: 250px;
top: 20 px;
width: 200px;
}
```

```
</style>
</head>
<body>
 Zawartość
pierwszego akapitu, ...
 Zawartość drugiego
akapitu, ...
</body>
</html>
```

Obecność i wygląd obramowania takich elementów jak tabele czy akapity można również kontrolować z wykorzystaniem kaskadowych arkuszy stylów. Służy do tego celu parametr border używany w połączeniu z określeniem strony elementu, której dotyczy definicja obramowania, np. border-left lub border-top.

Parametrowi należy przypisać wartość solid, aby określić, że obramowanie dla elementu ma być widoczne. Na przykład:

```
TABLE{ border-left: solid; border-
right: solid; border-top: solid; bor-
der-bottom: solid}
```

lub

```
TABLE {border: solid;}
```

Kolejnym parametrem określającym wygląd obramowania jest jego styl definiowany za pomocą parametru borderstyle.

Parametr boder-style może być określony dla całego elementu lub tylko dla niektórych jego stron, na przykład borderleft-style lub border-top-style. Parametr border-style może przyjmować następujące wartości:

- none, hidden ramka jest niewidoczna;
- dotted ramka składa się z linii wykropkowanej;
- solid ramka składa się z linii ciągłej;
- double ramka składa się z linii podwójnej;
- groove; ridge, outset ramka daje efekt wypukłości elementu;
- inset ramka daje efekt wklęsłości elementu.

Grubość obramowania elementów może mieć różną wartość i jest definiowana przez użycie parametru *border-width*. Również w przypadku tego parametru możemy definiować grubość krawędzi dla każdej strony elementu osobno, na przykład *border-bottom-width*.

Kolor obramowania jest definiowany przez użycie parametru border-color. Jeśli chcesz, aby każda z krawędzi elementu miała inny kolor, możesz zastosować następujący kod:

```
Ρ{
```

```
border-top-color: blue;
border-bottom-color: navy;
border-left-color: #FF0080;
border-right-color: #008080;
border-style: double;
```

}

Programy do tworzenia arkuszy stylów

Mam dla Ciebie dobrą wiadomość — nie zawsze będziesz musiał tworzyć kaskadowe arkusze stylów samodzielnie, ponieważ istnieją programy, które zrobią za Ciebie "brudną robotę". W takie funkcje wyposażone są dobre edytory WYSIWYG (jak na przykład Dreamweaver, o czym mówiliśmy szczegółowo w rozdziale 2.), ale programy do tworzenia arkuszy stylów możesz również pobrać z sieci. Do takich programów należy Style Master, którego testową wersję możesz pobrać pod adresem *http://www.westciv. com/style master*.



Jeśli zainteresowała Cię tematyka kaskadowych arkuszy stylów, możesz pogłębić swoją wiedzę, odwiedzając witryny poświęcone tematyce kaskadowych arkuszy stylów.

http://www.csszengarden.com



http://www.w3schools.com/css/



http://www.kurshtml.boo.pl/

